

# 蓝牙技术在机械手控制系统中的应用

张 粤 倪 伟 淮阴工学院(223001)

## Abstract

This paper which is based on the robot's hand control system discuss the framework and the working procedure which realize the bluetooth technological protocol. From application viewpoint, this paper introduces the realization of wireless communication between PC and microprocessor 78E58.

**Keywords:** robot's hand control system, bluetooth technology, communication principle, communication realize

## 摘 要

本文以机械手控制系统为背景,讨论了实现蓝牙技术协议的整体框架和主要的工作流程,并从应用的角度介绍了PC机与微处理器78E58芯片间无线通讯的实现。

**关键词:** 机械手控制系统, 蓝牙技术, 通讯原理, 通讯的实现

蓝牙 (Bluetooth) 技术是一种无线数据与语音通讯的开放性全球规范, 它以低成本的近距离无线链接为基础, 为固定与移动设备通讯环境建立了一个通用的无线电空中接口 (Radio Air Interface) 及其控制软件的公开标准。蓝牙设备工作在 2.4GHz 的工业、科学与医学频段上, 其特殊设计的射频部分为设备提供了快速、稳定、安全的连接, 此外, 为了进一步提高数据的抗干扰能力, 蓝牙技术采用了跳频扩展频谱技术, 为其在工业环境中推广应用奠定了技术基础。本文以机械手控制系统为背景, 讨论了实现蓝牙技术协议的整体框架和主要的工作流程, 从应用的角度介绍了PC机与微处理器78E58芯片间无线通讯的实现。

## 1 系统结构

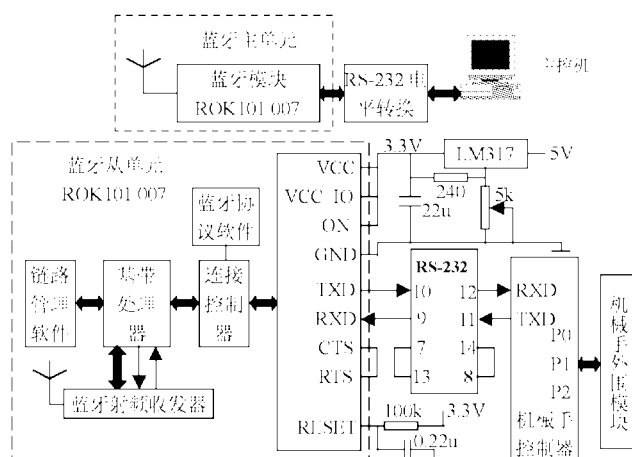


图1 系统结构

某企业产品包装线机械手控制系统由一个主控站(蓝牙无线基站)和八个远程站组成, 实现产品包装箱

的抓取、搬运、码放等功能。其工作过程为: 在主控站发出抓取产品箱体指令后, 远程站即控制机械手接近产品生产线, 到位后, 手部横向移入箱体下端, 接着手部上移, 直到使手部的压力传感器达到一级预设值为止。此时, 位于手部正上方的滚珠丝杠即推动吸盘组件下降, 压紧箱体使手部的压力传感器达到二级预设值。压紧后, 启动真空发生器, 吸盘吸紧。在抓取动作完成后, 机械手运动至指定码放位置, 手指回转, 吸盘松开, 箱体被码放到位。具体结构如图1所示。

蓝牙无线基站由主控机、蓝牙模块 ROK101 007 (主单元)、RS232 电平转换接口等组成。其中, ROK101 007 是由爱立信公司生产的集基带控制器和射频控制器于一体的点对多点蓝牙模块, 它除提供同步面向链接链路(SCO)和异步无链接链路(ACL)两种物理连接外, 还对外提供了 UART (传输数据)、PCM(传输音频信号)、USB(传输音频和数据)通讯接口, 在本系统中, 考虑到系统的通讯匹配问题及硬件开销, 我们采用 UART 接口实现主控机与主单元的通讯。此外, 主单元采用星型网络拓扑与八个远程蓝牙设备(从单元)进行射频链接, 形成 Piconet 网。在 Piconet 网内, 主单元以时段为单位与其它从单元建立 ACL 链接。在输入采样阶段, 主单元负责采集网内各从单元的现场数据, 经串口传输给主控机进行分析处理, 以协调各机械手的运行, 并实时刷新主控机的流程界面和控制界面; 在输出刷新阶段, 主单元接收主控机下达的控制指令及控制参数, 经 ACL 链路传输给 Piconet 网内相应的从单元, 以实现远程控制。

远程站由机械手控制器、蓝牙模块 ROK101 007、RS232 电平转换接口及其它外围器件组成。考虑到系统对传输速率及数据缓冲区的要求,机械手控制器由微处理器 78E58 实现,该芯片内嵌的增强型 8051 核除了提供片内程序存储器、片内额外的 256 字节 RAM 外,还为用户提供了编辑时钟周期的功能,从而为数据的高速传输奠定了硬件基础。微处理器 78E58 负责接收主控机下发的参数设定和控制指令,以控制机械手外围设备完成诸如手指定位、手指夹紧/放松、手指回转、手臂伸缩、手臂升降、底盘回转等动作,并将现场运行结果实时传递给主控机。

2 通讯原理

由于蓝牙设备的主控制器接口(HCI)封装了所有烦琐的技术细节,为用户提供了操作基带控制器和链路管理器的命令集,因此,在本系统中,我们通过 HCI 接口与远程主机建立无线连接,实现数据交互,具体通讯原理如图 2 所示。

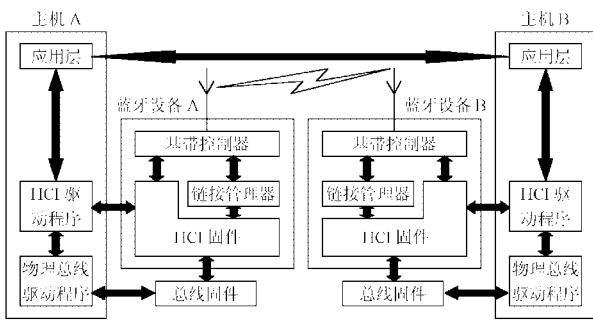


图 2 蓝牙通讯原理

2.1 建立物理信道

蓝牙设备 A 根据在查询过程中得到的设备 B 的基带地址(BD-ADD)确定它的呼叫跳频序列,并以此频率向设备 B 发送包含设备识别码(DAC)的 ID 分组,当处于呼叫扫描模式中的设备 B 发现呼叫请求为自身的设备识别码时,即在响应跳频序列中应答设备 A。当设备 A 接收到设备 B 的应答消息,即以同样的呼叫跳频序列向设备 B 发送一个 FHS 分组,此分组包含设备 A 的实时蓝牙时钟、48 位基带地址、BCH 奇偶校验位和设备类型。当设备 B 接收到 FHS 分组,即以自身设备识别码作为应答,以确认 FHS 分组的接收。设备 A 在接收到设备 B 的应答后,即发送 POLL 分组给设备 B,确认物理信道已连接完成。

2.2 建立 ACL 连接

在物理信道已建立的基础上,主机 A 根据得到的蓝牙设备 B 的地址(BD-ADD),通过 HCI 接口发送链接请求命令 HCI-Create-Connection (如图 3 所示)给蓝牙设备 A 的链路管理器(LM),在链接请求命令被 LM 成功接收后,它一方面将 LMP-Host-Con-

nection-Req 协议数据单元发送给基带控制器,经由无线物理信道传送给设备 B;另一方面,它向主机 A 回复一个命令状态事件(如图 4 所示),用于表明 LM 已接收到 HCI 命令,并正在进行处理。

操作码(2byte)		参数总长(1byte)	参数 0
OCF	OGF	参数……	
……			
参数 N-1		参数 N	

图 3 HCI 指令分组

事件码(1byte)	参数总长(1byte)	事件参数 0
事件参数 1	事件参数 2	事件参数 3
……		
事件参数 N-1	事件参数 N	

图 4 HCI 事件分组

当蓝牙设备 B 的基带控制器接收到设备 A 发来的 LMP-Host-Connection-Req 时,它即将请求传递给链路管理器(LM)。LM 识别出连接请求后,将自动产生一个链接请求事件,经 HCI 接口转发给主机 B,主机 B 通过发送 HCI-Accept-Connection-Request 或 HCI-Reject-Connection-Request 命令来决定是否接收主机 A 的链接请求。如果 LM 接收到链接接收命令 HCI-Accept-Connection-Request,则通知基带控制器传送 LMP-Accepted 协议数据单元给设备 A。

当蓝牙设备 A 的基带控制器接收到设备 B 发来的 LMP-Accepted 时,它即将应答消息传递给链路管理器(LM)。如果不需要进行鉴权和加密等安全性处理时,LM 就发送配置完成 LMP-Setup-Complete 协议数据单元给设备 B,当蓝牙设备 B 的 LM 成功接收此协议数据后,它一方面发送配置完成 LMP-Setup-Complete 协议数据单元给设备 A,同时向主机 B 回复一个包含链接句柄的链接完成事件,用于表明链接已经建立。同样,当蓝牙设备 A 的 LM 成功接收到来自设备 B 的 LMP-Setup-Complete 协议数据后,也向主机 A 回复一个包含相同链接句柄的链接完成事件,至此,主机 A 和 B 即可通过建立的 ACL 链路进行数据交互。

3 无线通讯的实现

根据蓝牙技术协议标准,通过 UART I/F,蓝牙模块 ROK101 007 能接收四种类型的 HCI 帧:指令分组、ACL 分组、SCO 分组、事件分组。同时,为了使蓝牙模块能够识别不同的 HCI 帧,蓝牙又规定在分组前

必须使用各自的分组指示,它们分别为 &H1、&H2、&H3、&H4。在本系统中,因没有涉及音频数据传输,在软件设计方面我们仅使用了三种 HCI 帧:指令分组、ACL 分组、事件分组。

此外,考虑到蓝牙通讯的特点,在主从机软件设计中,我们采用模块化设计、事件中断处理等方法,保证了各个模块间的并行工作,避免了数据处理之间的相互等待。

### 3.1 主控机初始化

在主控机通讯软件设计方面,我们基于 API 通讯函数,采用面向对象语言 VB 构造了一个驱动 HCI 接口的通讯类模块 HCIDriver,它封装和隐藏了 HCI 接口驱动程序的实现细节,并为用户提供了简明的调用函数。在主控机窗口加载模块中,我们首先调用 HCIDriver.CommOpen 函数创建通讯线程,在串口初始化工作完成后,主控机即激活线程阻塞函数 Wait,为接收主单元激活的事件分组和从单元传回的 ACL 数据分组作准备。其次,我们通过调用 HCIDriver.HCISend 函数发送 HCI-Host-Buffer-Size 命令,通知蓝牙主单元主机数据缓冲区的类型和大小,同样,用 HCI-Read-Buffer-Size 命令获取蓝牙主单元的数据缓冲区的类型和大小,以实现两者间的流量控制。在上述工作完成后,按系统要求主控机向主单元发送相应的 HCI 命令来设置主单元的链接参数,如设置主单元链路管理器等待来自从单元呼叫响应的最长时间、从单元接收链接的最长时间,是否只有指定设备类的设备才允许链接、是否自动识别链接、是否需要鉴权和加密等参数,具体流程如图 5 所示。

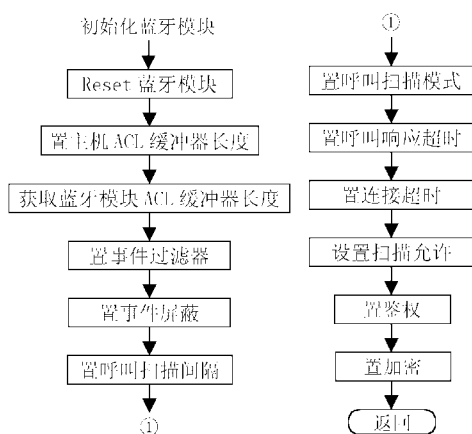


图 5 主控机初始化流程

串口初始化工作程序如下:

```

Private Sub InitComm()
Dim sec_att as SECURITY_ATTRIBUTES
Dim timeouts as COMMTIMEOUTS
Sec_att.nLength = Len(sec_att)
Sec_att.bInheritHandle = False
  
```

```

hdev = CreateFile ("com1", GENERIC_READ + GENER-
IC_WRITE, 0, sec_att, OPEN_EXISTING, FILE_AT-
TRIBUTE_NORMAL, 0)
If hdev = INVALID_HANDLE_VALUE Then
    success = MsgBox("error open com1", vbExclamation
+ vbOKOnly, "error windows")
Else
    success =SetupComm(hdev,1024,1024)
    pdcB.DCBLength = Len(DCB)
    success = GetCommState(hdev, pdcB)
    pdcB.BaudRate = 57600
    pdcB.ByteSize = 8
    pdcB.fBitFields = (pdcB.fBitFields Or &H1)
    pdcB.Parity = NOPARITY
    pdcB.StopBits = ONESTOPBIT
    success = SetCommState(hdev, pdcB)
    success = GetCommTimeouts(hdev, timeouts)
    timeouts.ReadIntervalTimeout = MAXWORD
    timeouts.ReadTotalTimeoutMultiplier = 0
    timeouts.ReadTotalTimeoutConstant = 0
    success = SetCommTimeouts(hdev, timeouts)
End If
End Sub
  
```

### 3.2 单片机初始化

与主控机初始化相类似,机械手控制器主程序首先初始化串口,如设置波特率为 57.6 Kb/s,选择工作方式 1 等。在串口初始化完成后,主程序按图 6 所示的流程初始化蓝牙从单元,使该模块处于查询扫描、呼叫扫描允许状态,以便在后续的链接建立进程中能正确响应主控机的 HCI-Inquiry、HCI-Create-Connection 命令请求,以完成主从机间的 ACL 链接,实现数据和控制指令的传递。

此外,根据蓝牙设备通讯的特点,在本系统中,机械手控制器主程序在主循环中采用非中断方式发送 HCI 命令分组和 ACL 数据分组,以响应主控机的请求。考虑到蓝牙从单元事件激活、与主控机数据交互的随机性,我们采用串口中断的方式接收 HCI 事件分组、HCI ACL 数据分组,并根据接收到的控制指令和控制参数调用相应的程序,完成相应的动作。

### 3.3 建立链接

在初始化完成后,主控机即发出 HCI-Inquiry 查询命令启动主单元的查询进程,主单元一旦在查询进程中接收到从单元的应答,即激活查询结果事件。在串行通讯模块接收到一个完整的查询结果事件分组后,阻塞函数 Wait 将激活 RecData 事件,通知主控机应用程序获取应答的从单元设备数以及与从单元设备相对应的 BD-ADD。随后,主控机发出 HCI-Create-

Connection 链接命令分别与对应的从单元建立 ACL 链接,考虑到传输数据的安全性,系统在数据链路层设置了鉴权和加密,在验证完成后,链接双方将密钥即链接字、链接句柄以链接字通知事件和链接完成事件通知双方主机,为将来传递 ACL 分组提供了技术保证,具体流程如图 6 所示。

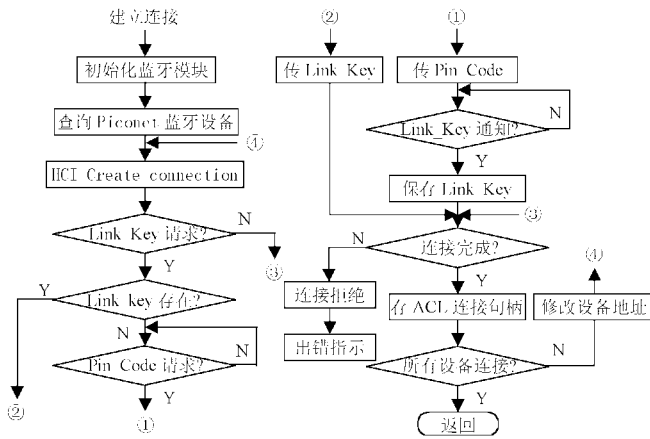


图 6 单片机初始化流程

阻塞函数程序如下:

```
Public Sub wait(byval hreodata as long)
Dim acceptbyte as byte
index = 0 : PacketTotalSize=0 '缓冲器指针及分组数据总长置零
Do While IOTerminated 'IOTerminated 置为 true 时运行 DoEvents
Do
success = ReadFile(hdev, acceptbyte, 1, returnbyte, my-overlapped)
If returnbyte = 1 Then
    accepchar(index) = acceptbyte
    index = index + 1
End If
Loop Until returnbyte = 0
If index >= 4 then
Select case accepchar(0)
    case is=4 '事件分组
        PacketTotalSize= accepchar(2)+3 '加事件分组头
    case is=2 'ACL 分组
        PacketTotalSize= accepchar(4) * 256
            + accepchar(3) + 5 '加 ACL 分组头
    case else
        '插入无效分组处理
End Select
If index >= PacketTotalSize Then
RaiseEvent recdata (accepchar, PacketTotalSize) '激活 recdata 事件
success = WaitForSingleObject(hreodata, INFINITE)
success = ResetEvent(hreodata) '数据接收后继续
```

```
j = 0
For i = PacketTotalSize To Index
accepchar(j)=accepchar(i):j=j+1
Next i
Index = Index-PacketTotalSize
End If
End If
Loop
End Sub
```

### 3.4 数据交互

在建立 ACL 链路的基础上,主控机与机械手控制器之间即可以 ACL 分组传递数据或系统自定义的控制数据,数据帧格式如图 7 所示。其中, PB 标志、BC 标志分别表示数据分组序号和数据传输类型,占两字节单元的数据总长是指后续有效载荷数据总的字节数亦即系统数据。



图 7 ACL 数据分组

在主机应用程序中,主控机通过调用 HCIDriver.HCISend 函数向机械手控制器传递控制指令及运动控制参数,以控制机械手的运动,如果在规定的时间内没有发送成功或没有接收到完成分组数事件则 HCIDriver.HCISend 返回出错代码,反之,HCIDriver.HCISend 将成功返回。

此外,为了使主控机应用程序能够完整的、准确的接收蓝牙主单元的事件分组以及机械手控制器传回的机械手现场数据,我们为 HCIDriver 定义了一个阻塞函数 Wait,当它接收到一个完整的事件分组(HCI 分组指示为 &H4)或 ACL 数据分组(HCI 分组指示为 &H2)时,将激活 HCIDriver 通讯模块的 RecData 事件,通知主控机窗口应用程序读取接收缓冲区中的数据,以更新窗口界面。

### 参考文献

- 1 Specification of the Bluetooth System.Specification Volume 1.BlueTooth,2001
- 2 Specification of the Bluetooth System.Specification Volume 2.BlueTooth,2001
- 3 ROK 101 007/1 Bluetooth Module,Ericsson,1999
- 4 金纯,许光辰,孙睿.蓝牙技术[M].北京:电子工业出版社,2001
- 5 Microsoft Corporation.ActiveX 控件参考手册.科学出版社,1998

[收稿日期:2002.8.15]